

```
*****
:
:           INTELLEC 4 MONITOR
:           MON4
:           VERSION 2.1
:           DECEMBER 1974
:
:           COPYRIGHT (C) 1974
:           INTEL CORPORATION
:           3065 BOWERS AVENUE
:           SANTA CLARA, CALIFORNIA 95051
:
:*****
```

ABSTRACT:

THIS IS THE MONITOR FOR THE INTELLEC 4 MICROCOMPUTER DEVELOPMENT SYSTEM. IT PROVIDES A SET OF COMMANDS BY WHICH THE OPERATOR CAN READ DATA FROM A PAPER TAPE, PUNCH THE CONTENTS OF RAM INTO PAPER TAPE, EXAMINE AND/OR MODIFY THE CONTENTS OF PROGRAM RAM, PROGRAM A 1602A OR 1702A PROM, AND LOAD THE CONTENTS OF A PROM INTO PROGRAM RAM.

ENVIRONMENT:

THE MONITOR IS DESIGNED TO BE USED WITH AN INTELLEC 4 CONNECTED TO AN ASR 33 TELETYPE, EQUIPPED WITH A PAPER TAPE READER/PUNCH. IN ADDITION, AN IMM4-90 HIGH-SPEED PAPER TAPE READER CAN BE USED IF ONE IS ATTACHED TO THE INTELLEC 4.

THE MONITOR IS DISTRIBUTED AS A SET OF FOUR 1702A PROMS, AND CURRENTLY REQUIRES 1019 8-BIT BYTES OF PROM MEMORY.

ERROR MESSAGES:

ERRORS ARE INDICATED BY AN ASTERISK (*).

THE FOLLOWING CONDITIONS WILL CAUSE AN ERROR:

1. INVALID COMMAND (SEE LIST BELOW)
2. INVALID CHARACTER IN NUMERICAL ENTRY
3. INVALID DELIMITER = NOT ' ', '.', ',', OR <CR>
4. CHECKSUM ERROR ON 'R' COMMAND
5. NOT READY CONDITION ON HIGH-SPEED READER.

6. MISCOMPARE DURING PROM PROGRAMMING

NOTE THAT CASES 4 AND 5 ARE INDISTINGUISHABLE WHEN READING A HEXADECIMAL TAPE FROM THE HIGH-SPEED READER. OPERATOR SHOULD BACK THE TAPE UP ABOUT 8 INCHES, CHECK FOR TEARS OR TAPE CAUGHT ON SOMETHING, AND RETYPE THE 'R' COMMAND. PERSISTENT FAILURE INDICATES CHECKSUM ERROR.

MISCOMPARES DURING PROM PROGRAMMING WILL CAUSE THE PROM ADDRESS AT WHICH THE MISCOMPARE OCCURRED TO BE PRINTED BEFORE THE ASTERISK,

PROGRAM ORGANIZATION

THE MONITOR IS ORGANIZED INTO FOUR DIFFERENT TYPES OF ROUTINES. A SHORT INITIALIZATION MODULE SETS THE TELETYPE PAPER TAPE READER AS THE DEFAULT INPUT DEVICE FOR TAPE, AND PREVENTS THE HIGH-SPEED READER (IF ATTACHED) FROM SLEWING TAPE. THE MAIN LOOP (LABEL 'START') ISSUES A PROMPT (.), AND LOOPS WAITING FOR KEYBOARD INPUT. WHEN A CHARACTER IS TYPED, IT IS DECODED AND USED AS AN INDEX INTO A JUMP TABLE. EACH COMMAND IS HANDLED BY A SEPARATE ROUTINE, WHICH MAY CALL ON SEVERAL UTILITY SUBROUTINES TO PERFORM SPECIFIC FUNCTIONS. IF THESE ROUTINES ENCOUNTER AN ERROR, THEY BRANCH TO 'LER', WHICH PRINTS AN ASTERISK AND RETURNS TO 'START'. NOTICE THAT THIS TAKES ADVANTAGE OF THE CIRCULAR SUBROUTINE STACK, SINCE COMPLETE ENTRY FROM SUBROUTINE NESTS IS NOT NECESSARY.

LIST OF VALID MONITOR COMMANDS

**** ** *****

- <LEGAL COMMAND> ::= <BNPF PUNCH COMMAND>
- | <DISPLAY MEMORY COMMAND>
- | <ENDFILE COMMAND>
- | <INVERT MEMORY COMMAND>
- | <ENABLE HIGH-SPEED PTP READER COMMAND>
- | <DISABLE HIGH-SPEED PTP READER COMMAND>
- | <LOAD BNPF COMMAND>
- | <MOVE MEMORY COMMAND>
- | <LEADER COMMAND>
- | <PROGRAM PROM COMMAND>

```

;           | <READ HEXADECIMAL FILE COMMAND>
;           | <SUBSTITUTE MEMORY COMMAND>
;           | <TRANSFER PROM COMMAND>
;           | <WRITE HEXADECIMAL RECORD COMMAND>
;
; <BNPF PUNCH COMMAND> ::= B<LOW ADDRESS>,<HIGH ADDRESS>
; <DISPLAY MEMORY COMMAND> ::= D<LOW ADDRESS>,<HIGH ADDRESS>
; <ENDFILE COMMAND> ::= E
; <INVERT MEMORY COMMAND> ::= I<LOW ADDRESS>,<HIGH ADDRESS>
; <ENABLE HIGH-SPEED PTP READER COMMAND> ::= J
; <DISABLE HIGH-SPEED PTP READER COMMAND> ::= K
; <LOAD BNPF COMMAND> ::= L<LOW ADDRESS>,<HIGH ADDRESS>
; <MOVE MEMORY COMMAND> ::= M<LOW ADDRESS>,<HIGH ADDRESS>,<DEST ADDRESS>
; <LEADER COMMAND> ::= N
; <PROGRAM PROM COMMAND> ::= P<LOW ADDRESS>,<HIGH ADDRESS>,<PROM ADDRESS>
; <READ HEXADECIMAL FILE COMMAND> ::= R
; <SUBSTITUTE MEMORY COMMAND> ::= S<ADDRESS>
; <TRANSFER PROM COMMAND> ::= T<ADDRESS>
; <WRITE HEXADECIMAL RECORD COMMAND> ::= W<LOW ADDRESS>,<HIGH ADDRESS>
;
; ALL ADDRESSES ARE SPECIFIED IN HEXADECIMAL,
; SYSTEM SIGNS ON WITH <CR><LF><,>
;
;*****
;
;*****
;
; GLOBAL DECLARATIONS
; TITLE ' INTELLEC 4 MONITOR, VERSION 2.1, 01 DEC 74 '
;
; EQUATED VARIABLES
```

```

;
00E0      WPMC EQU      14 SHL 4      ; RAM PORT USED TO CONTROL WPM FETCH
00F0      HICRD EQU     15 SHL 4      ; HIGH NIBBLE PORT FOR WPM
00E0      NIBS EQU     14 SHL 4      ; ADDRESS OF PORTS FOR WPM DATA (READ)
0001      WEN EQU      1              ; VALUE USED TO CONTROL WPM FETCH
0040      RCTL EQU     4 SHL 4        ; RAM PORT USED TO CONTROL TAPE READER
0001      RON EQU      1              ; READER ON COMMAND
000D      CR EQU       0DH           ; ASCII VALUE OF CARRIAGE RETURN
000A      LF EQU       0AH           ; ASCII VALUE OF LINE FEED
0020      SPACE EQU    20H           ; ASCII VALUE OF SPACE
002C      COMMA EQU    2CH           ; ASCII VALUE OF COMMA
003A      COLON EQU    3AH           ; ASCII VALUE OF COLON
003C      DLY EQU     60              ; 4.55 MS DELAY COUNTER
0000      RET EQU      0              ; NOMINAL RETURN VALUE
00D0      THREE EQU   13 SHL 4       ; CONSTANT FOR THREE PARAMETERS
0000      PRMAD EQU    0              ; PROM PROGRAMMING ADDRESS
0002      RDATA EQU    2              ; PROGRAMMING READ DATA PORT(S)
0002      PULSE EQU    0010B         ; 1702A PROGRAMMING VALUE
00C7      POLY EQU    -57 AND 255    ; 520 MS DELAY TIMER
0004      PTSAD EQU    4              ; ROM PORT FOR FAST PTR STATUS
0006      PTDAD EQU    6              ; ROM PORT FOR DATA FROM FAST PTR
00EA      CMDCTR EQU    100H-( 'W'-'B'+1) ; CONTROLS COMMAND LOOKUP
;

```

REGISTER DECLARATIONS

```

;
0000      R0 EQU      0
0001      R1 EQU      1
0002      R2 EQU      2              ; RESERVED FOR CHARACTER I/O
0003      R3 EQU      3              ; RESERVED FOR CHARACTER I/O
0004      R4 EQU      4
0005      R5 EQU      5
0006      R6 EQU      6
0007      R7 EQU      7
0008      R8 EQU      8
0009      R9 EQU      9
000A      RA EQU     10
000B      RB EQU     11
000C      RC EQU     12
000D      RD EQU     13
000E      RE EQU     14
000F      RF EQU     15
;

```

```

;*****
;
;
;
;*****
;
INITIALIZATION ROUTINE
;

```



```

001E 60      INC      R0
001F      MON3:  INC      R5      ; THIS CAN'T CAUSE AN OVERFLOW IF
001F 65      ; JUMP TABLE STARTS AT EVEN ADDRESS
; (WHICH IT DOES, BY FORCE)
; THIS MAY CAUSE OVERFLOW

0020 7523    ISZ      R5,MON4
0022 64      INC      R4
0023      MON4:
0023 7717    ISZ      R7,MON1    ; LOOP COUNTER - NEED BOTH TO
0025 7617    ISZ      R6,MON1    ; COUNT ABOVE 16
0027 40FA    JUN      LER      ; NOT FOUND - HE GOOFED

;
; COMMAND BRANCH TABLE - MUST BE ON EVEN ADDRESS
;
002A      ORG      8+(8 MOD 2)
002A      TBL:
002A 4286    JUN      BNPF      ; B - PUNCH BNPF
002C 40FA    JUN      LER      ; C -
002E 4200    JUN      DISP     ; D - DISPLAY RAM PROGRAM MEMORY
0030 4300    JUN      EOF      ; E - ENDFILE A HEXADECIMAL FILE
0032 40FA    JUN      LER      ; F -
0034 40FA    JUN      LER      ; G -
0036 40FA    JUN      LER      ; H -
0038 43E7    JUN      INVRT    ; I - INVERT MEMORY
003A 415B    JUN      PTRON    ; J - ENABLE HIGH-SPEED PTP READER
003C 4160    JUN      PTROF    ; K - DISABLE HIGH-SPEED PTP READER
003E 4375    JUN      LOAD     ; L - LOAD BNPF TAPE
0040 43AA    JUN      MOVE     ; M - MOVE MEMORY
0042 42BB    JUN      NULL     ; N - PUNCH NULLS FOR LEADER
0044 40FA    JUN      LER      ; O -
0046 421F    JUN      PROG     ; P - PROGRAM A PROM
0048 40FA    JUN      LER      ; Q -
004A 42C1    JUN      READ     ; R - READ HEXADECIMAL FILE
004C 4312    JUN      SUBS     ; S - SUBSTITUTE MEMORY
004E 43C8    JUN      TRANS    ; T - TRANSFER FROM PROM TO MEMORY
0050 40FA    JUN      LER      ; U -
0052 40FA    JUN      LER      ; V -
0054 41A8    JUN      WRITE    ; W - WRITE HEX TAPE

;
;
; *****
;
; MISCELLANEOUS ROUTINES
;
0056      CRLF:
0056 220D    FIM      R2,CR    ; <CR>
0058 5081    JMS      TO      ; TYPE OUT CHAR
005A      LFX:

```

```

005A 220A      FIM      R2,LF      ; <LF>
005C 4081      JUN      TO
;
; COMPARE POINTER AGAINST LIMIT
;
HILO:
005E          JMS      SRO      ; SELECT RAM CHIP 0, REGISTER 0
005E 53F7      RDO      ; ADDRESS BITS 11-8
0060 EC        XCH      RA
0061 BA        RD1      ; ADDRESS BITS 7-4
0062 ED        XCH      RB
0063 BB        RD2      ; ADDRESS BITS 3-0
0064 EE        XCH      RC
0065 BC        CLC
0066 F1        ISZ      RC,HIO    ; INCREMENT ADDRESS
0067 7C6E      ISZ      RB,HIO
0069 7B6E      ISZ      RA,HIO
006B 7A6E      BBL      RET
006D C0
HILO:
006E AA        LD      RA      ; REPLACE ADDRESS DATA
006F E4        WRO
0070 AB        LD      RB
0071 E5        WR1
0072 AC        LD      RC
0073 E6        WR2
0074 60        INC      RO
0075 21        SRC      RO      ; COMPARE LOW ADDRESS WITH HIGH ADDRESS
0076 EE        RD2
0077 9C        SUB      RC
0078 F3        CMC
0079 ED        RD1
007A 9B        SUB      RB
007B F3        CMC
007C EC        RDO
007D 9A        SUB      RA
007E C0        BBL      RET

```

```

;
; *****
;
; TYPE OUT (AND PUNCH) ONE SPACE.
;
; NOTICE THAT THIS PARAGRAPH FALLS INTO THE TELETYPE
; OUTPUT ROUTINE, 'TO', WHAT THE DESIGNER
; HATH JOINED TOGETHER, LET NO MAN PUT ASUNDER.
;
;

```

```

007F          BLK:      ; PUNCH A BLANK
007F 2220      FIM      R2,LF

```

L

TELETYPE OUTPUT ROUTINE

THIS ROUTINE EXPECTS A CHARACTER IN R2 & R3, WHICH IT WILL OUTPUT TO THE TELETYPE.

Reg 2, 3

```

0081
0081 53F7      TO1:  JMS   SRO      ; SELECT TTY PORT
0083 E1       WMP
0084 D8       LDM   0      ; START BIT
0085 B4       XCH   R4      ; R4 = BIT COUNTER
0086
0086 5359      TO1:  JMS   SBR2     ; 9.09 MS DELAY
0088 F1       CLC
0089 B2       XCH   R2      ; SHIFT 8 BITS (R2,R3) RIGHT
008A F6       RAR
008B B2       XCH   R2
008C B3       XCH   R3
008D F6       RAR
008E B3       XCH   R3
008F F7       TCC
0090 E1       WMP          ; CARRY -> ACC
0091 7486      ISZ   R4,TO1  ; OUTPUT BIT TO TTY
0093 5359      JMS   SBR2     ; CONTINUE LOOPING
0095 D1       LDM   1      ; 9.09 MS DELAY
0096 E1       WMP          ; STOP BIT 1
0097 5359      JMS   SBR2     ; 9.09 MS DELAY
0099 5359      JMS   SBR2     ; AND ANOTHER
009B C0       BDL   RET      ; RETURN
    
```

TELETYPE INPUT ROUTINE

READS CHARACTER FROM TELETYPE AND RETURNS IT IN R2, R3.

NOTE: ENTERING THIS ROUTINE AT 'TI' WILL CLEAR THE ACCUMULATOR AND CARRY BIT. ENTERING AT 'TIX' WILL LEAVE THE ACCUMULATOR AND CARRY UNDISTURBED.

```

009C
009C F0      TI:   CLB
009D
009D      TIX:
    
```


*this has to be
RAM 1, BIT 0*
*4 SHL 4
RAM Port Used To Control
TTY Reader*

009D 2640
 009F 27
 00A0 E1
 00A1 D8
 00A2 B4
 00A3 53F7
 00A5
 00A5 EA
 00A6 F6
 00A7 1AA5
 00A9 535F
 00AB 27
 00AC E1
 00AD 21
 00AE E1
 00AF
 00AF 5359
 00B1 EA
 00B2 F4
 00B3 E1
 00B4 F6
 00B5 A2
 00B6 F6
 00B7 B2
 00B8 A3
 00B9 F6
 00BA B3
 00BB 74AF
 00BD 5359
 00BF D1
 00C0 E1
 00C1 5359
 00C3 535F
 00C5 B2
 00C6 F5
 00C7 F1
 00C8 F6
 00C9 B2
 00CA C0

FIM R6, RCTL
 SRC R6 ; TWEAK READER
 WMP
 LDM 8
 XCH R4 ; R4 = BIT COUNTER
 JMS SRO (Fim 0,0 SRC 0)
 TI0:
 RDR *Lead Rom Input Port 0*
 RAR *BIT 0* ; SHIFT TO CARRY
 JNC TI0 ; LOOP WAITING FOR START BIT in bit 2010 position
 JMS SBR1 ; 4,55 MS DELAY
 SRC R6
 WMP
 SRC R0
 WMP ; ECHO START BIT
 TI1:
 JMS SBR2 ; 9,09 MS DELAY
 RDR ; INPUT DATA
 CMA ; COMPLEMENT ACC
 WMP ; ECHO DATA BIT
 RAR ; BIT TO LINK
 LD R2 ; GET UPPER NIBBLE
 RAR ; SHIFT IN CARRY
 XCH R2 ; SAVE UPPER NIBBLE
 LD R3 ; GET LOWER NIBBLE
 RAR ; SHIFT IN CARRY
 XCH R3 ; SAVE LOWER NIBBLE
 ISZ R4, TI1 ; GET ALL 8 BITS
 JMS SBR2 ; STOP BIT 1
 LDM 1
 WMP ; ECHO STOP BIT 1
 JMS SBR2
 JMS SBR1 ; 4,55 MS DELAY
 XCH R2 ; ELIMINATE PARITY BIT
 RAL
 CLC
 RAR
 XCH R2
 RET ; RETURN
 BBL

*its either a start
bit - then is not
ground.*

*ground
but
to pin 1
if reads
ground it
should not be
to 00B9*

;
;
;
;
;

CONVERT NIBBLE IN ACC TO ASCII IN R2,R3

00CB
 00CB 2230
 00CD F1

HXD:
 FIM R2, '0'
 CLC ; R2, R3 = 40
 ; set to
 ; ASCII EQUIVALENT
 ; REG PAIR TO 0

```

00CE   FB           DAA
00CF   1AD3         JNC   HX0
00D1   62           INC   R2
00D2   F2           IAC
00D3   B3           XCH   R3
00D4   4081         JUN   TO
;
;
;
;
;*****
;
;   PUNCH 60 NULL CHARACTERS (6 INCHES OF BLANK TAPE)
;
;   NOTE: THIS SUBROUTINE IS USED BY THE 'N' COMMAND
;   (SEE LABEL 'NULL'), AND BY THE 'P' COMMAND (SEE LABEL
;   'BNPF').
;
;LEAD:
00D6   2CC4         FIM   RC,-60 AND 255 ; SET COUNTER TO PUNCH 6 INCHES
00D8   2200         FIM   R2,0 ; PUNCH NULL
00DA   5081         LEO:  TO
00DC   7DDA         ISZ   RD,LEO
00DE   7CDA         ISZ   RC,LEO
00E0   C0           BBL   RET
;
;
;
;
;*****
;
;   DECODE AN ASCII CHARACTER IN REGISTERS R2, R3 INTO A SINGLE
;   HEXADECIMAL DIGIT IN REGISTER R2,
;
;   ANY CHARACTER OTHER THAN 0 - 9, A - F, IS INVALID,
;
;   EXIT WITH CARRY OFF; CHARACTER VALID
;   EXIT WITH CARRY ON; CHARACTER INVALID
;
;NIBBLE:
00E1   F0           CLB
00E1   D3           LDM   '0' SHR 4 ; TEST FOR 0-9
00E2   92           SUB   R2
00E3   1CED         JNZ   NI1 ; NO, TEST FOR A-F
00E4   F1           CLC
00E5   A3           LD    R3 ; NOW TEST LOWER 4 BITS
00E6   FB           DAA

```

```

00E9 12F8      JC      NI2      ; ILLEGAL CHARACTER
00EB          NI0:   XCH      R2      ; ACC -> R2 (RETURNED VALUE)
00EB B2        BBL      RET
00EC C0
00ED          NI1:   IAC
00ED F2
                                ; TEST FOR A-F, ASCII ZONE DIGIT FOR
                                ; THESE IS A 4, AND FOR DIGITS IT'S
                                ; A 3, THUS, IF IT WAS A-F, THE
                                ; ACCUMULATOR WILL CONTAIN 0FH.
                                ; ILLEGAL CHARACTER, ERROR RETURN

00EE 1CF8      JNZ      NI2
00F0 F1        CLC
00F1 D9        LDM      9
00F2 B3        XCH      R3
00F3 14F8      JZ       NI2      ; CHECK FOR 40 (H)
00F5 B3        ADD      R3
00F6 1AEB      JNC      NI0
00F8          NI2:   STC
00F8 FA        BBL      RET
00F9 C0

```

```

;
;
;
;
;
;
;*****
;

```

```

; ERROR ROUTINE - PRINT AN ASTERISK
;
00FA          LER:
00FA 222A      FIM      R2, '*' ; PRINT ERROR FLAG
00FC 5081      JMS      TO
00FE 4007      JUN      START

```

```

;
;
;
;
;
;*****
;

```

```

; THE FOLLOWING STATEMENTS SET THE ADDRESS OF P0 TO THE NUMBER
; OF BYTES LEFT ON THIS PAGE, THEN PAGE-ALIGNS THE ORIGIN.
;
0000          P0 EQU      (256-($ MOD 256)) AND 255
0100          ORG      $+P0

```

```

;
;
;
;
;

```

```

;*****
;
;   READ ONE CHARACTER FROM THE PAPER TAPE READER.
;
;   DATA RAM 0, REGISTER 0, STATUS CHARACTER 3 HOLDS THE
;   INPUT DEVICE ASSIGNMENT, IF THIS IS A '0', THE CHARACTER
;   IS READ FROM THE TELETYPE PTP READER, IN THIS CASE,
;   'ROM' CONTAINS THE CORRECT BIT PATTERN TO ADVANCE
;   THE TAPE FEED TO THE NEXT CHARACTER.
;
;   IF THE STATUS CHARACTER IS NOT ZERO, THE INPUT
;   CHARACTER IS READ FROM THE HIGH-SPEED PAPER TAPE READER.
;
0100          RTAPE:
0100      53F7      JMS      SR0          ; SET TO READ FROM RAM REGISTER 0
0102      EF        RD3          ; READ STATUS CHARACTER 3
0103      1C08     JNZ      RPTR       ; IF NOT ZERO, USE FAST READER
0105      D1        LDM      ROM       ; SET UP TO ADVANCE TAPE
0106      409D     JUN      TIX       ; GO TO TELETYPE READ ROUTINE
0108
0108      2240     RPTR:   FIM      R2,PTSAD SHL 4 ; THIS PUTS THE ROM PORT # FOR THE
; DATA AVAILABLE SIGNAL IN R2, AND A 0
; IN R3.
010A      23        SRC      R2
010B      D0        LDM      0          ; ADVANCE READER BY WRITING A 0 BIT TO
010C      E2        WRR      ; THE STATUS PORT
010D      D8        LDM      8          ; CLEAR READER ADVANCE IMMEDIATELY
010E      E2        WPR      ;
010F      2000     FIM      R0,0       ; R0 AND R1 ARE LOOP COUNTERS
0111
0111      EA        RPT0:  RDR          ; GET DATA AVAIL. SIGNAL
0112      F5        RAL          ; PUT BIT 3 IN CARRY
0113      1A1B     JNC      RPT1       ; IF BIT 3 IS OFF WE HAVE DATA
0115      7011     ISZ      R0,RPT0    ; KEEP TESTING AND WAITING,
0117      7111     ISZ      R1,RPT0    ; TESTING AND WAITING . . .
0119      40FA     JUN      LER       ; BROKEN TAPE
011B
011B      D6        RPT1:  LDM      PTAD  ; ROM PORT FOR LOWER 4 BITS OF DATA
011C      B4        XCH      R4        ; PUT IN R4
011D      25        SRC      R4
011E      EA        RDR          ; GET LOWER 4 BITS OF DATA
011F      F4        CMA          ; DATA COMES IN NEGATIVE LOGIC
0120      B3        XCH      R3        ; PUT IN R3
0121      64        INC      R4        ; ROM PORT FOR UPPER 4 BITS OF DATA
0122      25        SRC      R4
0123      EA        RDR          ; GET UPPER 4 BITS OF DATA
0124      F4        CMA          ; STRAIGHTEN IT OUT AGAIN
0125      F5        RAL          ; STRIP OFF PARITY BIT
0126      F1        CLC
0127      F6        RAR
0128      B2        XCH      R2        ; PUT IN R2

```

```

0129 C0          BBL      RET          ; RETURN
;
;
;
;*****
;
;          EVALUATE EXPRESSION
;
;          ENTER AT 'EXPR<N>' TO PICK UP AND STORE <N> HEX
;          NUMBERS. EACH NUMBER MAY BE FROM 1 TO 3 DIGITS LONG,
;          WITH EXTRA DIGITS RESULTING IN LEFT TRUNCATION. THE
;          NUMBERS WILL BE PUT INTO STATUS CHARACTERS 0,1,2 OF
;          DATA RAM CHIP 0, STARTING WITH REGISTER 0, THEN 1, ETC.
;          ADDITIONALLY, THE LAST NUMBER PICKED UP WILL BE LEFT IN
;          INDEX REGISTERS RD,RE,RF.
;
;          NUMBERS ARE DELIMITED BY COMMA OR CARRIAGE RETURN ONLY.
;
;          EXPR0:          ; ENTER HERE FOR ZERO EXPRESSIONS
;          EXPR1:          ; ENTER HERE FOR ONE EXPRESSION
;
012A          6C          INC          RC
;
012B          6C          EXPR2:       ; ENTER HERE FOR TWO EXPRESSIONS
;
012C          2A00       EXPR3:       ; ENTER HERE FOR THREE EXPRESSIONS
;
012E          2E00       EX0:         ; POINT TO FIRST RAM REGISTER
;
0130          D0         FIM          RA,0
;
0131          BD         FIM          ; ZERO OUT NUMBER COLLECTION BUFFER
;
;
;          EX1:          ; GET FIRST (NEXT) NUMBER
;
0132          509C       JMS          ; GET A CHARACTER
;
0134          50E1       JMS          ; TRANSLATE IT TO HEX
;
0136          123E       JC          ; NOT A HEX DIGIT
;
0138          A2         LD          ; SHIFT 12 BITS LEFT 4 PLACES
;
0139          BF         XCH          R2
;
013A          BE         XCH          RF
;
013B          BD         XCH          RE
;
013C          4132       XCH          RD
;
;          EX2:          ; GO BACK FOR ANOTHER DIGIT
;
013E          2B         SRC          ; CHECK FOR DELIMITER
;
013F          AD         LD          ; STORE NUMBER IN RD,RE,RF IN
;
0140          E4         WRO         ; STATUS CHARACTERS 0,1,2 OF
;
0141          AE         LD          ; CURRENT RAM REGISTER
;
;
0142          E5         WR1         RE
;
0143          AF         LD          RF
;
0144          E6         WR2
;
0145          6A         INC          RA
;
0146          202C       FIM          ; POINT TO NEXT REGISTER
;
0148          519C       JMS          ; CHECK FOR A COMMA
;
;          COMP

```

```

014A 1457      JZ      EX3      ; COMMA FOUND
014C 200D      FIM      RO,CR   ; CHECK FOR CARRIAGE RETURN
014E 519C      JMS      COMP
0150 1C59      JNZ      EX4      ; IF NOT CR IT'S BAD
0152 505A      JMS      LFX      ; CR = ADD A LINE FEED
0154 7C59      ISZ      RC,EX4   ; INSUFFICIENT OPERANDS
0156 C0        BBL      RET      ; ALL DONE
0157          EX3:
0157 7C2E      ISZ      RC,EX0   ; GO BACK IF WE NEED MORE
                                ; FALL THRU MEANS TOO MANY OPERANDS

0159          EX4:
0159 40FA      JUN      LER      ; SOME SORT OF ERROR

```

'J' AND 'K' COMMANDS - ENABLE AND DISABLE HIGH-SPEED PAPER TAPE READER, RESPECTIVELY.

NOTE: DATA RAM CHIP 0, REGISTER 0, STATUS CHARACTER 3 HOLDS THE STATUS OF THE READER:

 = 0 INPUT FROM TELETYPE TAPE READER
 = 1 INPUT FROM FAST TAPE READER

```

015B          PTRON:
015B 512A      JMS      EXPRO    ; WAIT FOR CR
015D D1        LDM      1        ; SET FOR FAST READER
015E 4163      JUN      JK        ; GO TO COMMON ROUTINE TO WRITE IT
0160          PTROF:
0160 512A      JMS      EXPRO    ; WAIT FOR CR
0162          PTRF2:
0162 D0        LDM      0        ; ENTRY PT FROM INITIALIZATION
0163          JK:
0163          ; SET FOR TELETYPE READER
0163 2000      FIM      RO,0     ; COMMON SUBROUTINE FOR THE ABOVE
0165 21        SRC      RO      ; SET UP REGISTERS FOR SRC
0166 E7        WR3      ; SELECT CORRECT RAM REGISTER
0167 4007      JUN      START   ; WRITE ACCUM. TO STATUS CHARACTER
                                ; RETURN TO MONITOR

```

STORE THE CONTENTS OF R2, R3 IN RAM PROGRAM MEMORY. THIS ROUTINE ASSUMES THAT THE ADDRESS IN WHICH THE DATA IS STORED IS LOCATED IN RAM STATUS CHARACTERS

```

; 0,1,2 OF CHIP 0, REGISTER 0.
;
STORE:
0169          FIM      R0,WPMC 14      ; TURN ON WRITE ENABLE
0169 20E0     SRC      R0
016B 21       LON      WEN 1
016C D1       WRR
016D E2       JMS      COMMON
016E 518E     LD        R2            ; HIGH NIBBLE OF DATA
0170 A2       WPM
0171 E3       LD        R3            ; LOW NIBBLE OF DATA
0172 A3       WPM
0173 E3       JMS      HILO          ; INCREMENT ADDRESS
0174 505E     BBL      RET 0
0176 C0

```

```

;
;
;
;
;*****
;

```

```

; FETCH A BYTE FROM RAM PROGRAM MEMORY.
; THIS ROUTINE ASSUMES THAT THE ADDRESS OF THE BYTE IS IN
; STATUS CHARACTERS 0,1,2 OF CHIP 0, REGISTER 0.
;
; RETURN ADDRESS OF BYTE IN REGISTERS RA, RB, RC.
; RETURN BYTE IN REGISTERS R2, R3.
;

```

```

0177          FIM      R0,WPMC
0177 20E0     SRC      R0
0179 21       CLB
017A F0       WRR          ; BLOCK WRITE ENABLE
017B E2       JMS      COMMON
017C 518E     WPM
017E E3       WPM
017F E3       FIM      R0,NIBS      ; ADDRESS OF DATA
0180 20E0     SRC      R0
0182 21       RDR
0183 EA       XCH      R2            ; HIGH NIBBLE OF DATA
0184 B2       INC      R0
0185 60       SRC      R0
0186 21       RDR
0187 EA       XCH      R3            ; LOW NIBBLE OF DATA
0188 B3       XCH      RB            ; ADJUST ADDRESS IN RA,RB,RC
0189 BB       XCH      RC
018A BC       XCH      RA
018B BA       XCH      RB
018C BB       XCH      RB
018D C0       BBL      RET

```

 COMMON SUBROUTINE USED BY 'FETCH' AND 'STORE'

```

COMMON:
018E      53F7      JMS      SR0 - 717 010
018E      ED        RD1          ; GET MIDDLE NIBBLE
0190      BA        XCH      RA          ; LOW NIBBLE
0191      EE        RD2          ; HIGH NIBBLE
0192      BB        XCH      RB          ; HIGH NIBBLE
0193      EC        RDO          ; ADDRESS OF HI-ORDER PORT
0194      20F0      FIM      RO,HIORD
0195      21        SRC      RO
0196      E2        WRR          ; SET UP A11-A8
0197      BC        XCH      RC
0198      2B        SRC      RA          ; SET A7-A0
0199      CO        BBL      RET
    
```

 COMPARE CHARACTER IN REGISTERS R0 AND R1 WITH CHARACTER IN
 REGISTERS R2 AND R3. COMP RETURNS RESULTS IN ACCUMULATOR:

ACCUM, ZERO CHARACTERS EQUAL
 ACCUM, ONE CHARACTERS UNEQUAL

ALSO, IF THE CHARACTERS ARE EQUAL, THE CARRY BIT WILL
 BE ON AT EXIT, IF THE CHARACTERS ARE UNEQUAL, THE SETTING OF THE
 CARRY BIT IS UNPREDICTABLE.

```

COMP:
019C      F1        CLC          ; RESET CARRY BIT
019C      A0        LD      R0          ; COMPARE HIGH-ORDER BITS
019D      92        SUB      R2
019E      1CA7      JNZ      UNEQL      ; HIGH-ORDER BITS UNEQUAL
019F      F1        CLC          ; RESET CARRY AGAIN
01A1      A1        LD      R1          ; COMPARE LOW-ORDER BITS
01A2      93        SUB      R3
01A3      1CA7      JNZ      UNEQL      ; LOW-ORDER BITS UNEQUAL
01A4      CO        BBL      0          ; RETURN WITH ZERO IN ACCUM.
01A6      C0
01A7      UNEQL:   BBL      1          ; RETURN WITH 1 IN ACCUM.
01A7      C1
    
```



```

;
;
;*****
;
; 'W' COMMAND - WRITE A HEXADECIMAL TAPE FROM RAM MEMORY
;
WRITE:
01A8      512B      JMS      EXPR2      ; GET 2 ADDRESSES
01AA      5056      WRIO:    JMS      CRLF      ; PUT RECORD ON A NEW LINE
01AC      223A      FIM      R2,';'
01AE      5081      JMS      TO
01B0      2800      FIM      R8,00      ; INITIALIZE CHECKSUM
01B2      5177      JMS      FETCH      ; GET ADDRESS IN (RA,RB,RC)
01B4      2210      FIM      R2,16      ; SET NOMINAL RECORD LENGTH
01B6      2610      FIM      R6,10H      ; SET NOMINAL COUNT AND TERMINATION FLAG.
01B8      F1        CLC
01B9      AF        LD      RF      ; SUBTRACT LOW-ORDER DIGIT OF
01BA      9C        SUB      RC      ; CURRENT ADDRESS (IN RC) FROM
01BB      BC        XCH      RC      ; LOW-ORDER DIGIT OF END ADDRESS
; (IN RF) AND PUT RESULT BACK IN
; RC, THIS SETS THE CARRY BIT, WHICH
; MUST BE LEFT THIS WAY FOR THE
; REST OF THE SUBTRACTION, WHICH
; HAPPENS BELOW.
; ADD 16 TO THE CURRENT ADDRESS
01BC      7BC2      ISZ      RB,WRI1      ; IF THIS OVERFLOWS, WE'RE AT THE TOP
01BE      7AC2      ISZ      RA,WRI1      ; OF MEMORY, AND THIS IS THE LAST RECORD
01C0      41CA      JUN      WRI2
01C2      WRIO:    CMC      ; THIS FINISHES THE SUBTRACTION WE
01C2      F3        LD      RE      ; STARTED ABOVE, WE'RE NOT INTERESTED
01C3      AE        SUB      RB      ; IN THE DIFFERENCE, ONLY THE SETTING
01C4      9B        CMC      ; OF THE CARRY BIT AT THE END.
01C5      F3        LD      RD
01C6      AD        SUB      RA
01C7      9A        JC      WRI4
01C8      12D4      WRIO:    CLB
01CA      F0        XCH      R2      ; LAST RECORD, <= 16 BYTES
01CB      B2        LD      RC
01CC      AC        XCH      R3
01CD      B3        ISZ      R3,WRI3      ; INCREMENT R2,R3
01CE      73D1      INC      R2
01D0      62        WRIO:    LD      RC
01D1      AC        CMA
01D1      AC        XCH      R7      ; SET LOOP COUNT FOR LAST RECORD
01D2      F4        WRIO:    JMS      PBYTE      ; PUNCH RECORD LENGTH
01D3      B7        JMS      'FETCH      ; GET LOAD ADDRESS IN (RA,RB,RC)
01D4      5366      CLB
01D4      5366      JMS      'FETCH
01D6      5177      CLB
01D8      F0

```

```

01D9 B2 XCH R2
01DA BA XCH RA
01DB B3 XCH R3
01DC 5366 JMS PBYTE ; PUNCH FIRST BYTE OF LOAD ADDRESS
01DE BB XCH RB
01DF B2 XCH R2
01E0 BC XCH RC
01E1 B3 XCH R3
01E2 5366 JMS PBYTE ; PUNCH SECOND BYTE OF LOAD ADDRESS
01E4 2200 FIM R2,0
01E6 5366 JMS PBYTE ; PUNCH RECORD TYPE
01E8 WRIS:
01E8 5177 JMS FETCH ; GET A DATA BYTE
01EA 5366 JMS PBYTE ; PUNCH IT
01EC 505E JMS HILO ; INCREMENT & TEST COMPLETION
01EE F7 TCC ; SAVE CARRY BIT AS LAST-RECORD FLAG
01EF B6 XCH R6
01F0 77E8 ISZ R7,WRIS
01F2 F0 CLB
01F3 99 SUB R9 ; NEGATE CHECKSUM
01F4 B3 XCH R3
01F5 F3 CMC
01F6 D0 LDM 0
01F7 98 SUB R8
01F8 B2 XCH R2
01F9 5366 JMS PBYTE ; PUNCH CHECKSUM
01FB A6 LD ; TEST FOR COMPLETION
01FC 1CAA WRIO ; PUNCH ANOTHER RECORD
01FE 4007 JUN START

```

```

;
;
;
;
;*****
;

```

FOLLOWING STATEMENTS SET THE ADDRESS OF 'P1' TO THE NUMBER OF BYTES LEFT ON THIS PAGE, THEN PAGE-ALIGNS THE ORIGIN.

```

0000 P1 EQU (256-(8 MOD 256)) AND 255
0200 ORG 8+P1

```

16 16

```

;*****
;

```

'D' COMMAND = DISPLAY MEMORY IN HEX ON TELETYPE

```

0200 DISP:

```

2

254

256

256

100

109

100

100


```

0225 25 SRC R4
0226 AF LD RF ; LOWER 4 BITS OF FROM ADDRESS
0227 E2 WRR
0228 64 INC R4
0229 25 SRC R4
022A AE LD RE ; UPPER 4 BITS OF FROM ADDRESS
022B E2 WRR
022C 5177 JMS FETCH ; GET A BYTE OF DATA
022E 64 INC R4
022F 25 SRC R4
0230 A3 LD R3
0231 F4 CMA
0232 E2 WRR ; LOWER 4 BITS OF DATA
0233 64 INC R4
0234 25 SRC R4
0235 A2 LD R2
0236 F4 CMA
0237 E2 WRR ; UPPER 4 BITS OF DATA
0238 2640 FIM R6,RCTL
023A 27 SRC R6
023B D2 LDM PULSE ; TURN ON PROGRAMMER
023C E1 WMP
023D 28C7 FIM R8,PDLY ; 520 MS DELAY
023F PR2:
023F 5359 JMS SBR2 ; 9,09 MS AT A TIME
0241 793F ISZ R9,PR2
0243 783F ISZ R8,PR2
0245 F0 CLB ; TURN OFF PROGRAMMER
0246 E1 WMP
0247 5359 JMS SBR2 ; 9 MS FOR SETTLING AND COOLING
0249 D2 LDM RDATA ; PUT ADDRESS OF PROM READ PORT
024A B4 XCH R4 ; IN R4
024B 25 SRC R4
024C EA RDR
024D F4 CMA
024E B1 XCH R1 ; PUT LOWER 4 BITS FROM PROM IN R1
024F 64 INC R4
0250 25 SRC R4
0251 EA RDR
0252 F4 CMA
0253 B0 XCH R0 ; PUT UPPER 4 BITS FROM PROM IN R0
0254 519C JMS COMP ; COMPARE R0,R1 WITH R2,R3
0256 1464 JZ PR4 ; ACCUM, ZERO MEANS COMPARE OK
0258 7D23 ISZ RD,PR1 ; ABORT AFTER 3 TRIES
025A 507F JMS BLK ; TYPE A SPACE
025C AE LD RE
025D B2 XCH R2
025E AF LD RF
025F B3 XCH R3
0260 5366 JMS PBYTE ; TYPE ADDRESS OF ERROR
0262 40FA JUN LER

```

```

0264          PR4:          ISZ    RF,PR5          ; INCREMENT PROM ADDRESS
0264    7F6A          ISZ    RE,PR5
0266    7E6A          JUN     START              ; HIT PROM ADDRESS 00
0268    4007
026A          PR5:          JMS    HILO          ; INCREMENT RAM ADDRESS
026A    505E          JC     PRO
026C    1221          JUN     START              ; ALL DONE
026E    4007

;
;
;
;
;*****
;
;   READ TWO ASCII CHARACTERS AND DECODE INTO ONE 8-BIT BYTE.
;
;   EXAMPLE:  '42' -> 'B', '3A' -> '!', ETC.
;
;   NOTE:  ERRONEOUS INPUT (SUCH AS 'WG', '76', ETC.) WILL
;   BE DISCOVERED IN ROUTINE 'NIBBLE'.
;
; BYTE:
0270          JMS    RTAPE          ; GET FIRST CHARACTER
0270    5100          JMS    NIBBLE         ; DECODE INTO HEX DIGIT
0272    50E1          XCH    R2              ; SAVE CONTENTS OF R2 IN RF
0274    B2           XCH    RF
0275    BF           JMS    RTAPE          ; GET SECOND CHARACTER
0276    5100          JMS    NIBBLE         ; DECODE INTO HEX DIGIT
0278    50E1          XCH    R2              ; MOVE R2 TO R3 (LOWER DIGIT)
027A    B2           XCH    R3
027B    B3           XCH    RF              ; MOVE RF TO R2 (UPPER DIGIT)
027C    BF           XCH    R2
027D    B2           CLC
027E    F1           LD     R9              ; ACCUMULATE CHECKSUM
027F    A9           ADD    R3
0280    B3           XCH    R9
0281    B9           LD     R8
0282    A8           ADD    R2
0283    B2           XCH    R8
0284    B8           BBL    RET
0285    C0

;
;
;
;
;*****
;
;   'B' COMMAND - ROUTINE TO PUNCH A BNPF TAPE
;
;   PUNCH CONTENTS OF RAM MEMORY BETWEEN TWO NUMBERS

```

```

      )
      )
      ) ENTERED AT THE KEYBOARD
      )
      ) BPNF:
0286       )
0286       512B       ) JMS     EXPR2       ) ; FETCH 2 NUMBERS FROM THE KBD
0288       50D6       ) JMS     LEAD        ) ; PUNCH LEADER
028A       )
028A       DC        ) LDM     12         )
028B       B6        ) XCH     R6         ) ; SET COUNTER FOR 4 WORDS/LINE
028C       )
028C       5177      ) JMS     FETCH       ) ; GET THE DATA
      )
      )
      ) ENCODE THE CONTENTS OF REGISTERS R2, R3 AND PUNCH THEM IN
      ) BPNF FORM.
      )
      )
      ) NOTE: THE BPNF OUTPUT IS PUNCHED IN COMPLEMENTED
      ) (N=LOGIC) FORM.
      )
028E       B2        ) XCH     R2
028F       BA        ) XCH     RA
0290       B3        ) XCH     R3
0291       BB        ) XCH     RB
0292       D8        ) LDM     8           ) ; SET COUNTER
0293       B9        ) XCH     R9
0294       2242      ) FIM     R2,'B'      ) ; PUNCH A 'B'
0296       5081      ) JMS     TO
0298       )
0298       F1        ) CLC
0299       BB        ) XCH     RB
029A       F5        ) RAL
029B       BB        ) XCH     RB
029C       BA        ) XCH     RA
029D       F5        ) RAL
029E       BA        ) XCH     RA
029F       12A5      ) JC      BN4
02A1       2250      ) FIM     R2,'P'      ) ; PUNCH A 'P'
02A3       42A7      ) JUN     BN5
02A5       )
02A5       224E      ) FIM     R2,'N'      ) ; PUNCH AN 'N'
02A7       )
02A7       5081      ) JMS     TO
02A9       7998      ) ISZ     R9,BN3
      )
      )
      ) DONE WITH THIS CHARACTER - PUNCH AN 'F' TO FINISH IT
      )
02AB       2246      ) FIM     R2,'F'      ) ; PUNCH AN 'F'
02AD       5081      ) JMS     TO
02AF       507F      ) JMS     BLK
02B1       505E      ) JMS     HILO       ) ; INCREMENT AND TEST FOR COMPLETE
02B3       1ABD      ) JNC     NU0
02B5       768C      ) ISZ     R6,BN1     ) ; CONTINUE WITH THIS LINE
02B7       5056      ) JMS     CRLF

```

```

02B9 428A      JUN      BNO      ; START A NEW PRINT LINE
;
;
;
;*****
; 'N' COMMAND = PUNCH LEADER/TRAILER TAPE
;
02BB          NULL:
02BB 512A      JMS      EXPRO      ; WAIT FOR CR
02BD          NU0:   JMS      LEAD      ; ENTRY POINT FOR OTHER ROUTINES
02BD 50D6      JMS      LEAD
02BF 4007      JUN      START
;
;
;
;*****
; 'R' COMMAND = READ A HEXADECIMAL TAPE INTO RAM PROGRAM MEMORY
;
02C1          READ:
02C1 512A      JMS      EXPRO      ; WAIT FOR CR
02C3          RED0:
02C3 5100      JMS      RTAPE      ; GET A CHARACTER FROM TAPE
02C5 203A      FIM      R0,COLON  ; CHECK FOR RECORD MARK (COLON)
02C7 519C      JMS      COMP
02C9 1CC3      JNZ      RED0      ; NOT COLON = KEEP LOOKING
02CB 2800      FIM      R8,0      ; CLEAR CHECKSUM
02CD 5270      JMS      BYTE      ; GET COUNT
02CF A3        LD      R3
02D0 1CD5      JNZ      RED1
02D2 A2        LD      R2
02D3 14FE      JZ      RED4      ; ZERO RECORD LENGTH, ALL DONE
02D5          RED1:
02D5 F0        CLB
02D6 93        SUB      R3      ; NEGATE COUNT, PUT IN RD,RE
02D7 BE        XCH      RE
02D8 F3        CMC
02D9 D0        LDM      0
02DA 92        SUB      R2
02DB BD        XCH      RD
02DC 5270      JMS      BYTE      ; BYTE 0 OF ADDRESS
02DE 53F7      JMS      SRO
02E0 A3        LD      R3
02E1 E4        WR0
02E2 5270      JMS      BYTE      ; PUT HIGH ORDER NIBBLE IN RAM
02E4 53F7      JMS      SRO      ; BYTE 1 OF ADDRESS

```

```

02E6 A2 LD R2
02E7 E5 WR1
02E8 B3 XCH R3
02E9 E6 WR2 ; PUT LOWER BYTE IN RAM
02EA 5270 JMS BYTE ; READ RECORD TYPE
02EC RED2: JMS BYTE ; GET A DATA BYTE
02EE 5169 JMS STORE ; PUT IN MEMORY, INCREMENT POINTER
02F0 7EEC ISZ RE,RED2 ; INCREMENT COUNT
02F2 7DEC ISZ RD,RED2
02F4 5270 JMS BYTE ; READ CHECKSUM
02F6 A8 LD R8
02F7 1CFC JNZ RED3 ; CHECKSUM ERROR
02F9 A9 LD R9
02FA 14C3 JZ RED0
02FC RED3: JUN LER ; LOAD ERROR, PRINT '*'
02FE RED4: JUN START
02FE 4007 JUN START
;
;
;
;
;*****
;
; FOLLOWING STATEMENTS SET THE ADDRESS OF 'P2' EQUAL TO THE NUMBER
; OF BYTES LEFT ON THIS PAGE, THEN PAGE-ALIGNS THE ORIGIN,
;
0000 P2 EQU (256-($ MOD 256)) AND 255
0300 ORG $+P2
;
;
;
;*****
;
; 'E' COMMAND - PUNCH AN END OF FILE ('100') INTO HEX TAPE
;
0300 EDF: JMS EXPRO ; WAIT FOR CR
0300 512A FIM R2,';' ; PUNCH A COLON
0302 223A TO
0304 5081 JMS TO ; PUNCH ZEROES FOR RECORD LENGTH
0306 2230 FIM R2,'0'
0308 5081 JMS TO
030A 2230 FIM R2,'0'
030C 5081 JMS TO
030E 5056 JMS CRLF ; END WITH <CR><LF>
0310 42BD JUN NUO ; AND 60 NULLS FOR TRAILER
;

```



```

;
;
;
;
;*****
;
;      'S' COMMAND = MODIFY RAM PROGRAM MEMORY WITH KEYBOARD INPUTS
;
SUBS:
0312      512A      JMS      EXPR1      ; GET STARTING ADDR FROM KBD
0314      A3       LD       R3          ;
0314      5177      JMS      FETCH      ; FETCH A BYTE FROM PROGRAM RAM
0316      A3       LD       R3          ; SAVE R3 IN R9
0317      B9       XCH      R9          ;
0318      A2       LD       R2          ; CONVERT FIRST DIGIT TO ASCII
0319      50CB     JMS      HXD        ; AND DISPLAY
031B      A9       LD       R9          ; SAME FOR SECOND DIGIT
031C      50CB     JMS      HXD        ;
031E      222D     FIM      R2,'-'      ; FOLLOW WITH A '-'
0320      5081     JMS      TO          ;

;
0322      2E00     FIM      RE,0        ; RE,RF WILL HOLD SUBSTITUTE DATA
0324      2C00     FIM      RC,0        ; RD IS SUBSTITUTE/IGNORE FLAG
0326      509C     SU1:   JMS      TI          ; GET A CHARACTER FROM TELETYPE
0328      50E1     JMS      NIBBLE     ; CONVERT TO HEX DIGIT IN R2
032A      1233     JC       SU2        ; CARRY ON IF NOT DIGIT
032C      2C0F     FIM      RC,OFH     ; SET SUBSTITUTE FLAG IN RD
032E      A2       LD       R2          ; MOVE R2 TO RF
032F      BF      XCH      RF          ; AND RF TO RE
0330      BE      XCH      RE          ;
0331      4326     JUN      SU1        ; GO BACK FOR NEXT DIGIT
0333      SU2:   ; DELIMITER FOUND
0333      2020     FIM      RO,SPACE    ; TEST FOR SPACE
0335      519C     JMS      COMP
0337      1C47     JNZ      SU4
0339      7D43     ISZ      RD,SU3     ; SPACE FOUND - DO WE STORE ANYTHING?
033B      AF      LD       RF          ; YES - MOVE RE,RF TO R2,R3
033C      B3      XCH      R3
033D      AE      LD       RE
033E      B2      XCH      R2
033F      5169     JMS      STORE      ; PUT NEW DATA IN RAM
0341      4314     JUN      SU0        ; GO BACK FOR NEXT LOCATION
0343      SU3:   ;
0343      505E     JMS      HILO      ; ADVANCE ADDRESS WITHOUT STORING
0345      4314     JUN      SU0        ; GO BACK FOR NEXT LOCATION
0347      SU4:   ;
0347      200D     FIM      RO,CR      ; WAS IT A CARRIAGE RETURN?
0349      519C     JMS      COMP
034B      144F     JZ       SU5
034D      40FA     JUN      LER        ; YES - CONTINUE
; NOT CARRIAGE RETURN - ERROR

```

```

034F      7D57      S05:      ISZ      RD,S06      ; DO WE STORE ANYTHING?
0351      AF        LD        RF        ; YES - MOVE RE,RF TO R2,R3
0352      B3        XCH      R3
0353      AE        LD        RE
0354      B2        XCH      R2
0355      5169      JMS      STORE      ; STORE THE DATA
0357      4007      S06:      JUN      START
;
;
;
;
;*****
;
;      TIMING DELAY SUBROUTINE
;
;      ENTER AT 'SBR2' FOR 9.09 MS DELAY (WHOLE BIT FROM TTY)
;      ENTER AT 'SBR1' FOR 4.55 MS DELAY (HALF BIT FROM TTY)
;
0359      203C      SBR2:      FIM      R0,DLY      ; 9.09 MS DELAY
0359      203C      L2:        FIM      R0,DLY
035B      715B      ISZ      R1,L2
035D      705B      ISZ      R0,L2
;
035F      203C      SBR1:      FIM      R0,DLY      ; 4.55 MS DELAY
035F      203C      L1:        FIM      R0,DLY
0361      7161      ISZ      R1,L1
0363      7061      ISZ      R0,L1
0365      C0        BBL      RET
;
;
;
;*****
;
;      PUNCH A BYTE AS TWO ASCII CHARACTERS
;
0366      F1        PBYTE:    CLC          ; ACCUMULATE CHECKSUM IN R8, R9
0367      A3        LD        R3
0368      89        ADD      R9
0369      B9        XCH      R9
036A      A2        LD        R2
036B      88        ADD      R8
036C      B8        XCH      R8
036D      B3        XCH      R3          ; SAVE R3 IN R5

```

*Toll Free #
000-538-8014
15*

```

036E B5 XCH R5
036F A2 LD R2 ; CONVERT UPPER NIBBLE
0370 50CB JMS HXD
0372 A5 LD R5 ; CONVERT LOWER NIBBLE
0373 40CB JUN HXD

;
;
;
;*****
;
; 'L' COMMAND = LOAD A BNPF TAPE INTO RAM PROGRAM MEMORY.
;
; NOTE: THIS ROUTINE ASSUMES THAT THE BNPF TAPE IS
; PUNCHED IN NEGATIVE LOGIC; I.E., A 'P' WILL BE
; STORED AS A '0' AND AN 'N' WILL BE STORED AS A '1'.
;
LOAD:
0375 512B JMS EXPR2 ; GET 2 ADDRESSES FROM KEYBOARD
0377 L01: JMS RTAPE ; GET A CHARACTER FROM TAPE
0377 5100 FIM R0,'B' ; CHECK FOR A 'B'
0379 2042 JMS COMP
037B 519C JMS COMP
037D 1C77 JNZ L01 ; NOT A 'B' - TRY AGAIN
037F D8 LDM S ; SET UP LOOP COUNTER
0380 B9 XCH R9 ; IN R9
0381 2A00 FIM RA,0 ; RA,RB = 0
0383 L02: JMS RTAPE ; GET A CHARACTER
0383 5100 FIM R0,'P' ; CHECK FOR A 'P'
0385 2050 JMS COMP
0387 519C JMS COMP
0389 1C9C JNZ L04 ; NOT 'P'
038B F1 CLC ; SHIFT IN A ZERO (COMPLEMENT TAPE)
038C L03: XCH RB ; LOWER NIBBLE OF BYTE
038C BB RAL ; SHIFT IT
038D F5 XCH RB ; PUT IT BACK
038E BB XCH RA ; UPPER NIBBLE OF BYTE
038F BA RAL ; SHIFT IT (WITH CARRY)
0390 F5 XCH RA ; PUT IT BACK
0391 BA ISZ R9,L02 ; CONTINUE LOOPING
0392 7983 JMS RTAPE
0394 5100 JMS COMP
0396 BA XCH RA
0397 B2 XCH R2
0398 BB XCH RB
0399 B3 XCH R3
039A 43A4 JUN L06 ; CHARACTER COMPLETE
039C L04: FIM R0,'N' ; TEST FOR AN 'N'
039C 204E JMS COMP ; COMP RETURNS CARRY ON IF CHARS =
039E 519C

```

```

03A0 148C      JZ      LO3      ; 'N' FOUND - PUT IN ANOTHER BIT
03A2 40FA      JUN      LER      ; CHARACTER NOT N, P, OR F
03A4          LO6:    JMS      STORE    ; PUT CHAR IN MEMORY, INCREMENT POINTER
03A4 5169      JC      LO1      ; NOT DONE YET = CONTINUE
03A6 1277      JUN      START    ; DONE = RETURN TO MONITOR
03A8 4007
;
;
;
;
;

```

```

;*****
;
; 'M' COMMAND - MOVE A BLOCK OF RAM PROGRAM MEMORY FROM ONE
; LOCATION TO ANOTHER.
;
;

```

```

03AA          MOVE:  JMS      EXPR3    ; GET 3 NUMBERS FROM KEYBOARD
03AA 512C      MVO:   JMS      FETCH    ; GET A BYTE FROM PROGRAM RAM
03AC          ;
03AC 5177
;

```

```

; THE FOLLOWING CODE DUPLICATES THE ACTIONS OF ROUTINE 'STORE'.
; 'STORE' IS NOT USED, BECAUSE THE CODE NECESSARY TO SAVE
; REGISTERS AND DO THE SETUP IS LONGER THAN THE CODE BELOW.
;
;

```

```

03AE 20E0      FIM      RO,WPMC
03B0 21        SRC      RO
03B1 D1         LDM      WEN
03B2 E2        WRR
03B3 20F0      FIM      RO,HIORD
03B5 21        SRC      RO
03B6 AD        LD       RD
03B7 E2        WRR
03B8 2F        SRC      RE
03B9 A2        LD       R2
03BA E3        WPM
03BB A3        LD       R3
03BC E3        WPM
;

```

```

03BD 7FC2      ISZ      RF,MV1    ; ADD 1 TO A 12-BIT NUMBER
03BF 7EC2      ISZ      RE,MV1
03C1 6D        INC      RD
03C2          MVO:

```

```

03C2 505E      JMS      HILO    ; INCREMENT HIGH ADDRESS AND TEST
03C4 12AC      JC       MVO     ; KEEP GOING
03C6 4007      JUN      START   ; ALL DONE
;
;
;
;

```

```

;*****
;
; 'T' COMMAND - TRANSFER CONTENTS OF PROM TO PROGRAM RAM
;
TRANS:
03C8          512A      JMS     EXPR1      ; GET 1 NUMBER FROM KEYBOARD
03C9          2800      FIM     R8,0
03CC          D0       TRO:   LDM     PRMAD     ; PUT ADDRESS OF PROM PORT IN R4
03CD          B4       XCH     R4
03CE          25       SRC     R4
03CF          A9       LD      R9           ; LOWER 4 BITS OF ADDRESS
03D0          E2       WRR
03D1          64       INC     R4
03D2          25       SRC     R4
03D3          A8       LD      R8           ; UPPER 4 BITS OF ADDRESS
03D4          E2       WRR
03D5          64       INC     R4
03D6          25       SRC     R4
03D7          EA       RDR
03D8          F4       CMA
03D9          B3       XCH     R3           ; LOWER 4 BITS OF DATA
03DA          64       INC     R4
C3DB          25       SRC     R4
03DC          EA       RDR
03DD          F4       CMA
03DE          B2       XCH     R2           ; UPPER 4 BITS OF DATA
03DF          5169      JMS     STORE     ; PUT BYTE IN RAM
03E1          79CC      ISZ     R9,TRO     ; INCREMENT PROM ADDRESS
03E3          78CC      ISZ     R8,TRO
03E5          4007      JUN

```

```

;
;
;*****
;
; 'I' COMMAND - INVERT MEMORY
;
INVRT:
03E7          512B      JMS     EXPR2      ; GET LOW & HIGH ADDRESSES
03E9          5177      INVO:  JMS     FETCH     ; GET A BYTE FROM PROGRAM RAM
03EB          A2       LD      R2           ; GET HIGH-ORDER DIGIT
03EC          F4       CMA     ; COMPLEMENT IT
03ED          B2       XCH     R2           ; AND REPLACE IT
03EE          A3       LD      R3           ; SAME FOR LOW-ORDER DIGIT
03EF          F4       CMA
03F0          B3       XCH     R3
03F1          5169      JMS     STORE     ; REPLACE IT IN PROGRAM RAM

```

03F3 12E9 JC INVO ; NOT DONE YET - KEEP GOING
03F5 4007 JUN START ; ALL DONE

;
;
;
;

; SRO JUST SETS UP PORTS (ROM OR RAM) 0 FOR I/O.

03F7 SRO: FIM R0,0
03F7 2000 SRC R0
03F9 21 BBL RET
03FA C0

;
;
;
;

; FOLLOWING LINE SETS ADDRESS OF P3 TO THE NUMBER OF BYTES LEFT
; ON THIS PAGE.

0005 P3 EQU (256-(\$ MOD 256)) AND 255
END

NO PROGRAM ERRORS

SYMBOL TABLE

* 01

BLK	007F	BN0	028A	BN1	028C	BN3	0298
BN4	02A5	BN5	02A7	BNPF	0286	BYTE	0270
CMDCT	00EA	COLON	003A	COMMA	002C	COMMO	018E
COMP	019C	CR	000D	CRLF	0056	DIO	0202
DI1	020C	DI2	021D	DISP	0200	DLY	003C
EOF	0300	EX0	012E	EX1	0132	EX2	013E
EX3	0157	EX4	0159	EXPRO	012A	EXPR1	012A
EXPR2	012B	EXPR3	012C	FETCH	0177	HIO	006E
HILO	005E	HIOFD	00F0	HX0	00D3	HXD	00CB
INIT	0000 *	INVO	03E9	INVRT	03E7	JK	0163
L1	0361	L2	035B	LE0	00DA	LEAD	00D6
LER	00FA	LF	000A	LFX	005A	LO1	0377
LO2	0383	LO3	038C	LO4	039C	LO6	03A4
LOAD	0375	MON1	0017	MON2	001C	MON3	001F
MON4	0023	MOVE	03AA	MVO	03AC	MV1	03C2
NIO	00EB	N11	00ED	N12	00F8	NIBBL	00E1
NIBS	00E0	NU0	02BD	NULL	02BB	PO	0000
P1	0000	P2	0000	P3	0005 *	PBYTE	0366
PDLY	00C7	PR0	0221	PR1	0223	PR2	023F
PR4	0264	PR5	026A	PRMAD	0000	PROG	021F
PTDAD	0006	PTRF2	0162	PTRCF	0160	PTRON	015B
PTSAD	0004	PULSE	0002	R0	0000	R1	0001
R2	0002	R3	0003	R4	0004	R5	0005
R6	0006	R7	0007	R8	0008	R9	0009
RA	000A	RB	000B	RC	000C	RCTI	0040
RD	000D	RDATA	0002	RE	000E	READ	02C1
REDO	02C3	RED1	02D5	RED2	02EC	RED3	02FC
RED4	02FE	RET	0000	RF	000F	RON	0001
RPT0	0111	RPT1	011B	RPTR	0108	RTAPE	0100
SBR1	035F	SBR2	0359	SPACE	0020	SRO	03F7
START	0007	STORE	0169	SU0	0314	SU1	0326
SU2	0333	SU3	0343	SU4	0347	SU5	034F
SU6	0357	SUBS	0312	TBL	002A	THREE	00D0
TI	009C	TIO	00A5	TI1	00AF	TIX	009D
TO	0081	TO1	0086	TRO	03CC	TRANS	03C8
UNEQL	01A7	WEN	0001	WPMC	00E0	WRIO	01AA
WRI1	01C2	WRI2	01CA	WRI3	01D1	WRI4	01D4
WRI5	01E8	WRITE	01A8				